Problem A. Automat

Input file:	`automat.in`
Output file:	`automat.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

A *nondeterministic finite automaton without ϵ-transitions* is a directed graph, the nodes of which are called *states*, and the edges of which are called *transitions*. Each transition has an assigned *label*, which in this problem will always be either 0 or 1. Exactly one state is the *starting state*, and some subset of states, possibly including the starting state, is called the *final states*.

A *word* is a sequence of labels, in our case of 0s and 1s. An automaton is said to be *accepting* a given word if and only if there's a path from the starting state to one of the final states such that the labels in sequence along this path comprise the word.

Given a word of length n, find any nondeterministic finite automaton without ϵ-transitions that has at most $\lfloor \frac{n}{2} \rfloor + 1$ states such that it accepts the given word, but no other word of length n. $\lfloor x \rfloor$ denotes the largest integer not exceeding x.

Input

The first line of the input file contains the number of testcases t. Each of the next t lines contains one word of 0s and 1s. The length of each word is at least 1 and at most 100, the total length of all words in one input file is at most 10^5. All words in one input file are distinct.

Output

For each word, output the description of an automaton that accepts the given word but no other word of the same length, or a single line with -1 on it if there's no such automaton.

The first line of an automaton description should contain four numbers k, m, s and t, denoting the number of states, the number of transitions, the identifier of the starting state, and the number of final states. The states have identifiers from 1 to k. The second line should contain t distinct identifiers of the final states. The next m lines should contain the transitions, each transition described by three numbers: the identifier of the source state, the identifier of the destination state, and the label. There should be no duplicate transitions (but there might be two transitions that differ only by the label). k must not exceed $\lfloor \frac{len(word)}{2} \rfloor + 1$.

All numbers in each line should be separated by spaces.

Examples

automat.in	automat.out
2 0 01	1 1 1 1 1 1 1 0 2 2 1 1 2 1 2 0 2 2 1